

AD-A096 099

NAVAL RESEARCH LAB WASHINGTON DC F/6 9/2  
A FORTRAN IV PROGRAM TO SCALE AND PLOT LOGARITHMIC AND LINEAR D--ETC(U)  
MAR 81 T R O'NEAL

**F/6 9/2**

UNCLASSIFIED

NRL-MR-4480

NL

1 OF 1  
20 A  
096094

END  
DATE  
FILMED  
4-81  
DTIC

AD A U 90099

9 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4480 ✓	2. GOVT ACCESSION NO. AD-A096 099	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  A FORTRAN IV PROGRAM TO SCALE AND PLOT LOGARITHMIC AND LINEAR DATA		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  T. R. O'Neal		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS  RR013-06-45; 61-0027-0-1
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 19, 1981
		13. NUMBER OF PAGES 45
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Hewlett-Packard                      Logarithm Graphics                                  Fortran Scaling Plotting		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report provides a general plotting routine to express 2 dimensional tabular data in logarithmic or linear graphical form. The program contains a scaling routine to maximize plot coverage and calculate axis length and tick marks. The routines are written in Fortran IV computer language and designed to run on a Hewlett-Packard HP 1000 computer system with Graphics 1000 plotting software and a graphics terminal or plotter.		

## TABLE OF CONTENTS

Introduction .....	1
Features of Program .....	2
Computer and System Configuration .....	3
User Changes .....	3
Program Background .....	3
Cautions .....	4
Input .....	5
Loading Procedure on HP 1000 .....	5
Program Source Availability .....	6
Error Messages .....	6
Testing of Program .....	6
Execution of Program .....	6
Subroutines .....	7
Plotter Set up .....	8
Plotting Accuracy .....	9
Mnemonic List .....	10
Formulas .....	12
Flow Charts .....	13
Examples .....	16
References .....	24
Sample Dialogue of Program Execution .....	25
Program Listings .....	26

11 t Special  
 A

## A FORTRAN IV PROGRAM TO SCALE AND PLOT LOGARITHMIC AND LINEAR DATA

### Introduction

The need to represent data in a form more meaningful than tabular has prompted the development of new software for general scientific use. Our specific needs in chemical experimentation require the capability for plotting of a wide variety of X vs. Y data types, with and without "error bar"-type error limits on each data point. In addition, it is often necessary to do least squares calculations on these same data and plot a least squares regression line on the same plot for visual indication of linearity, scatter and goodness-of-fit. Finally, the capability to provide journal-ready plots to eliminate the need for the user to make decisions about scaling is highly desirable.

Program LPLOT satisfies these basic requirements and contains such additional capabilities as log, semi-log or linear axes, multiple data sets on a single graph, multi-colored plots and variable origin starting location. The main routine does the plotting; five subroutines handle the log and linear scaling, log and linear axis drawing and calculation of a least squares regression. All six routines are written in Fortran IV. <sup>(1)</sup> This software was developed to provide maximum flexibility with a minimum of effort by the user. It is designed to run on a Hewlett-Packard 1000 computer system under an RTE IV operating system and makes use of the device-independent features of Hewlett-Packard's Graphics-1000 Software as well as some H.P. extensions to standard Fortran IV.

The most desirable feature of program LPLOT is that the calls to external routines are only to control plotter pen motion, this makes the program easily adaptable to run with any graphic package.

This document is designed to serve the purposes of a users' guide and an operations manual, and to provide sufficient documentation for program maintenance.

Manuscript submitted January 19, 1981.

### Features of Program

1. Automatic linear and logarithmic scaling.
2. Choice of linear, semi-log, or logarithmic axes.
3. Least squares line with slope, Y-intercept and standard deviations of each.
4. Error bars.
5. Multiple plots with different symbols on same graph.
6. Data source can be from disc file, cartridge or paper tape or typed in from the keyboard of a terminal.
7. Output can be on graphics terminal or plotter.
8. Plots can be line, symbols, or symbols connected by line.
9. Axis labeling and title are entered from terminal.
10. Origin can be determined automatically or forced to start at (0,0).
11. Tick marks are labeled.
12. Multiple color plots with automatic pen changing.
13. A manual scaling option can plot multiple data sets, which have different maximum and minimum values, on a single set of axes.
14. The routine can handle up to 256 points per data set.

### Computer and System Configuration

This program could be modified to run on many different computers with different plot packages; however, this routine was written for the Hewlett-Packard HP 1000 family of computers using Graphics-1000 (HP 92840A graphics plotting software). The operating system is RTE IVB with updated software revision code 2001. A graphics device, HP 2648A graphics terminal and/or HP 9872B plotter, is needed to do the plotting. Older versions of the RTE operating system and earlier plotters can be used so long as graphics 1000 limitations are satisfied. EMA and spooling features were not utilized. This routine requires a 26K-word partition to run.

### User Changes

This program was tested on a system probably configured differently from that of the user. It is the responsibility of the user to make changes to logical unit assignments to implement this program on his system. Changes will need to be made to lines 39 through 44, 49, 299 and 390 of program LPLOT. LUG is the logical unit number of the graphics device. ID is the identification number assigned by the device link table to a graphics logical device. Line 68 of program LPLOT does not permit a logical unit number greater than 30. This limit was set to avoid input errors and may need to be changed by the user.

### Program Background

The Hewlett-Packard Graphics 1000 plot package (HP 92840A) does not provide logarithmic plotting, linear or logarithmic scaling. It is left for the user to determine the limits of the data and how the axes of a plot are to be annotated.

The existing graphics software forces the user to draw two linear axes with one call to an axes drawing routine. This action restricts both axes annotation to one format. Program LPLOT is not limited by this restriction because it uses two calls, one for each axis, so each axis can be annotated differently.

The need to represent data logarithmically as opposed to linearly is evident when data spans many orders of magnitude or conforms to logarithmic interpretation.

It is because of these needs and restrictions program LPLOT was written.

### Cautions

Erroneous control codes in a data file or failure to separate data with a comma or space can produce unpredictable plotting results.

Data files set up for error bar plotting must set DELX and DELY equal to zero for those data points where error bars are not desired.

The subroutine SCALE transforms the array A into exponential notation, storing the exponent in IEXP. This process changes the array A by some factor of 10. If the user calls this subroutine from any other program he should be aware that any value passed back to the calling routine could be changed. It is for this reason A and IEXP must be used together.

A symbol type plot cannot center a symbol directly on an axis because the plotting window boundary is set on the axis and calls cannot be made to plot outside this area.

Some errors in input data can generate error conditions in Hewlett-Packard library routines. As written, the program allows these to be printed on the standard list device (logical unit 6). To avoid this, the user must supply his own error routine as described in relocatable library routine ERO.E. (2)

The user should be aware that significance in single precision real format is to six or seven decimal digits. It is due to this limitation that plotting errors will be encountered if input data exceeds this limit.

## Input

Data can be input to the program from a disc file, cartridge or paper tape or typed in from the keyboard of a terminal. Disc files must be type 3 or 4.<sup>(3)</sup> The program can handle an array up to 256 data points. The format is X, Y, DELX, DELY using free field input.<sup>(4)</sup> The X and Y pair is the position of the point along the X and Y axis respectively. The optional pair DELX and DELY is the standard deviation in X and Y. These values are used for drawing error bars and will be doubled and scaled to plotter units to provide the horizontal or vertical separation of error bars. All numeric input data must be in the range of  $10^{-38}$  to  $10^{38}$ .

## Loading Procedure on HP 1000

The loader must be loaded as a large background program (type 4) and will require a size of 26 pages to accommodate the Graphics-1000 routines. After compiling &LPLOT, &SCALE, &LSCAL, &LINAX, &LOGAX and &LSREG, execute the loader on the command file or interactively as follows:

RU,LOADR	RE,%LINAX
SZ,26	RE,%LOGAX
OP,LB	RE,%LSREG
RE,%LPLOT	RE,%DLTBL
RE,%SCALE	SE,%GPS
RE,%LSCAL	END

where %DLTBL is the device link table and %GPS is the Graphics-1000 library file created when Graphics-1000 was loaded.<sup>(5)</sup> Alternatively, the source version available from the author contains a loader command file which can be used. The loaded program will occupy approximately 26K words of memory. It would be possible to decrease the main program space somewhat by using EMA<sup>(6)</sup> for array space. However, the bulk of this 26K words is required for Graphics-1000 routines.

### Program Source Availability

A program source is available from the author on a user-supplied Hewlett-Packard 264X-type cartridge tape. This tape contains 8 files. The first file is a description of what is on the tape. The second file is a command file which may be used by the loader to load the programs. The third file is the source for program LPLOT. The fourth file is the source for subroutine SCALE. The fifth file is the source for subroutine LSCAL. The sixth file is the source for subroutine LINAX. The seventh file is the source for subroutine LOGAX. The eighth file is the source for subroutine LSREG.

### Error Messages

Program LPLOT checks for errors that could occur when reading a disc resident file. Messages to the user are sent with the name of the file if a problem is encountered. The program has several built in checks to catch typing errors by the user.

### Testing of Program

The author has used a wide variety of data types, multiple files, and several input sources to debug this program. In addition, the program has been used extensively by other individuals with different applications.

### Execution of Program

The execution of LPLOT can be performed by supplying terminal and graphics device Logical Unit Numbers (LU and LUG) via the run string parameters. These parameters (globals) are retrieved by a call to the Hewlett-Packard routine RMPAR in LPLOT. If these parameters are not supplied, the program retrieves LU and prompts the user for the graphics device LUG. Device selection is to be made between the plotter, for a hard copy, or the graphics terminal. The data source is from a disc file, cartridge or paper

tape or entered via the keyboard of a terminal. If the source is a disc file the name of the file is requested. If the source is other than the disc a logical unit number is requested. One of three types of plots can be selected; a straight line connecting each point, a symbol at each point or a symbol at each point with connecting lines. The user may select either axis to be linear or logarithmic. Labels are entered from the users terminal, any ASCII character is permitted (capital and lower case letters, numbers and symbols). The X and Y axis labels cannot exceed 30 characters and the title of the plot cannot exceed 40 characters in length; characters beyond these limits are ignored. The starting position of the origin can be forced to start at  $X = 0$  and  $Y = 0$  or the user can let the scale routine determine an origin that will maximize the size of the plot vs. the size of the axes. Scaling is performed automatically, however a manual override is provided. The manual scaling mode is used to increase the limits between the maximum and minimum values of an axis. This feature is necessary when plotting multiple plots on one graph when the maximum and minimum values of all of the data sets are not within one data set. When plotting multiple data sets the first set plotted must have the smallest and largest values of all of the data sets. If this condition cannot be met the user must specify manual scaling and enter minimum and maximum values of the entire set of data. A least squares line can be drawn on the plot with slope, y-intercept and respective standard deviations printed on the users terminal or printer. Error bars can be drawn around each point provided requirements in the input section of this paper have been met. Error bars that are small enough to distort the symbol printed at a data point are suppressed and a message is printed on the users terminal of this action. Six or less plots can be made using the same set of axes provided all data sets fall within the limits of the first data set plotted. The user has the option to make pen color changes when doing multiple plots on the same axes.

### Subroutines

The axis drawing and scaling subroutines of program LPLOT do not make calls to external routines other than for plotter pen movements.

The subroutine SCALE uses a table look-up method, based on the difference between maximum and minimum values, to determine axis scaling and number of tick marks to be placed on each axis. The SCALE routine uses an algorithm that shifts decimal points to increase numbers that are less than one and decrease numbers that are greater than 1000.

The subroutine LSCAL logarithmically scales an array to determine the exponent to which the base 10 must be raised to produce a number less than or equal to the minimum value of the array and greater than or equal to the maximum value of the array.

The subroutine LINAX draws a linear axis and labels tick marks in the X or Y direction.

The subroutine LOGAX draws a logarithmic axis and labels tick marks in the X or Y direction.

The subroutine LSREG does a least squares regression calculation including the standard deviation of the slope and y intercept.

#### Plotter Set Up

The HP 9872B plotter is used when hardcopy results of LPLOT are required. The plot generated by LPLOT is designed to fit on standard 8½ X 11 inch paper with adequate margins for publication. In order to center the plot the paper must be placed horizontally ¼ inch from the left side and 1 inch from the bottom of the plotter bed. This displacement is necessary because the HP 9872B plotter with advance option OFF places the lower left corner of the plotting window at (520, 380) instead of (0,0). For multi-colored plots pen placement is as follows; pen 1 is black, 2 is red, 3 is green, and 4 is blue. If the plot is to be one color that color pen must be in pen holder 1.

### Plotting Accuracy

The HP 9872B plotter is divided into plotter units where one unit = 0.025 mm. This is to say that the overall resolution of the plot is one part in N where N is the number of plotter units occupied in the X or Y direction. The X axis is 7.7 inches (7,823 plotter units) and the Y axis is 5.25 inches (5,334 plotter units). The plotter resolution is one part in 7,823 in the X direction and one part in 5,334 in the Y direction.

### Mnemonic List

BARX - Minimum value of X error bar  
BARY - Minimum value of Y error bar  
DELX - Experimental error in X (standard deviation)  
DELY - Experimental error in Y (standard deviation)  
DIF - Difference between maximum and minimum  
HH - Half height in character cells  
HW - Half width in character cells  
IAX - Label of X axis (30 characters max)  
IAXES - Type of plot (log and/or linear)  
IAY - Label of Y axis (30 characters max)  
IBAR - Type of error bars  
ID - Identification number  
IDONE - Check for termination or multiple plot  
IEXP - Exponent of base 10 in X data scale  
IFMT - Source of data  
IHED - Title of plot (40 characters max)  
ILINE - Check for least squares line  
IPEN - Pen number of plotter  
ISCAL - Set to zero for automatic scaling, 1 for manual  
IZERO - Set to zero to force origin to start at zero  
JCHAR - Character to be plotted at each data point  
JEXP - Exponent of base 10 in Y data scale  
JJ - Type of plot  
KK - Number of plots on same axes  
LU - Logical unit  
LUG - Logical unit of graphics device  
LUT - Logical unit of cartridge tape or keyboard  
NAME - Name of data file  
NOBAR - Check to see if error bar was too small to plot  
NP - Number of points  
S1 - Standard deviation of slope  
S2 - Standard deviation of Y intercept

SLOPE - Slope of least squares line  
SXTIC - Interval between X tick marks  
SYTIC - Interval between Y tick marks  
X - Displacement along X axis  
XBAR1 - Distance to left of character of X error bar  
XBAR2 - Distance to right of character of X error bar  
XEND - X value at end of least squares line  
XMAX - Maximum value of X  
XMIN - Minimum value of X  
XST - X value at start of least squares line  
XTIC - Number of tick marks on X axis  
Y - Displacement along Y axis  
YBAR1 - Distance above character of Y error bar  
YBAR2 - Distance below character of Y error bar  
YEND - Y value at end of least squares line  
YINT - Y intercept of least squares line  
YMAX - Maximum value of Y  
YMIN - Minimum value of Y  
YST - Y value at start of least squares line  
YTIC - Number of tick marks on Y axis

## Formulas

The following formulas were used in subroutine LSREG:

$$\text{Slope} = \frac{(\text{DIF})(\Sigma X \Sigma Y) - (\Sigma X \Sigma Y)}{(\text{DIF})(\Sigma X)^2 - (\Sigma X)^2}$$

$$\text{Y-Intercept} = \frac{\Sigma Y - (\text{SLOPE})(\Sigma X)}{\text{DIF}}$$

$$\text{Standard Deviation of Slope} = \sqrt{\frac{Y^2 - \frac{(\Sigma Y)^2}{\text{DIF}} - \frac{\left( (\Sigma X \Sigma Y) - \frac{\Sigma X \Sigma Y}{\text{DIF}} \right)^2}{\Sigma X^2 - \frac{(\Sigma X)^2}{\text{DIF}}}}{(\text{DIF} - 2)(\Sigma X^2) - \frac{(\Sigma X)^2}{\text{DIF}}}}$$

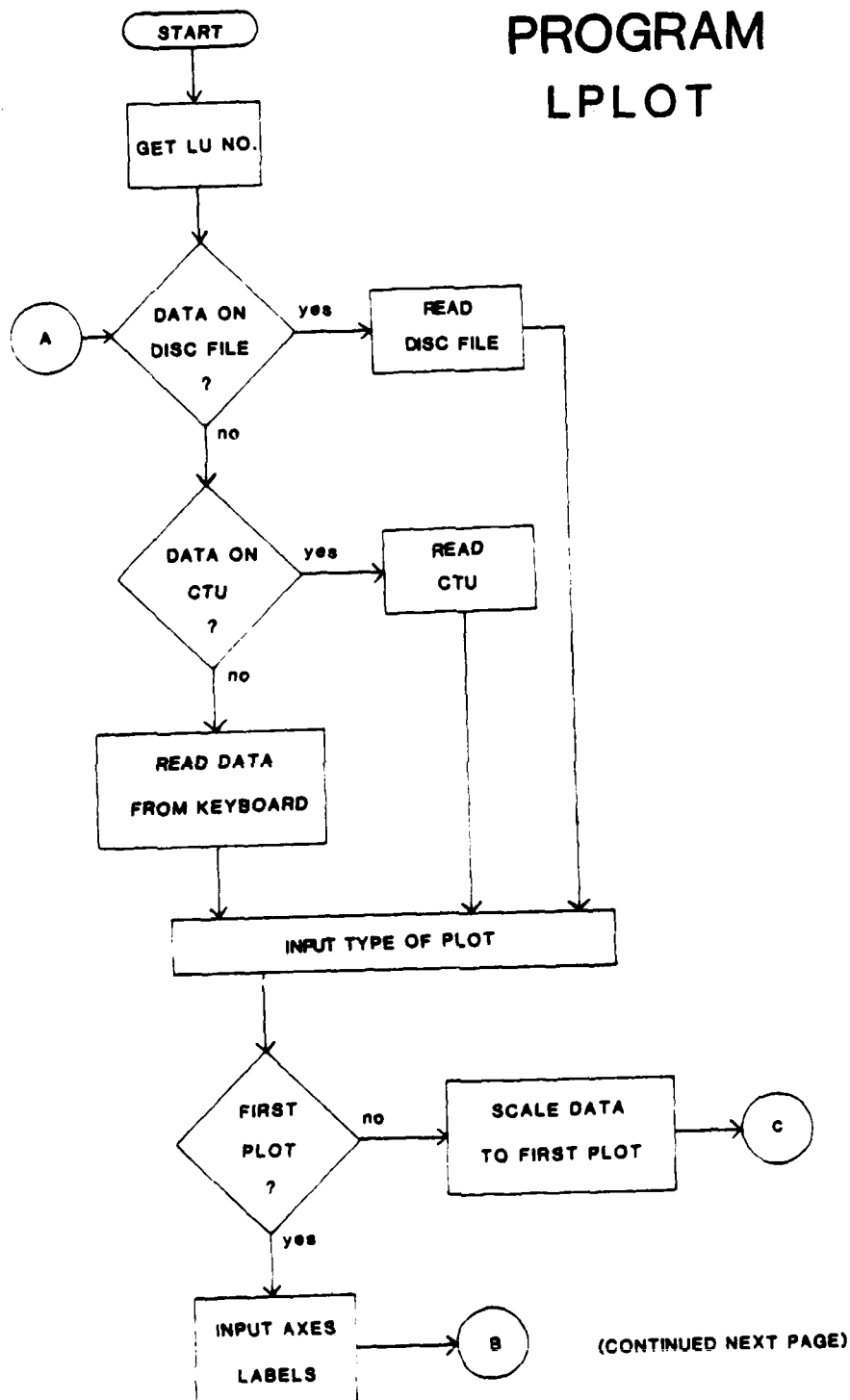
$$\text{Standard Deviation of Y Intercept} = \sqrt{(\text{SD of SLOPE})^2 \frac{\Sigma X^2}{\text{DIF}}}$$

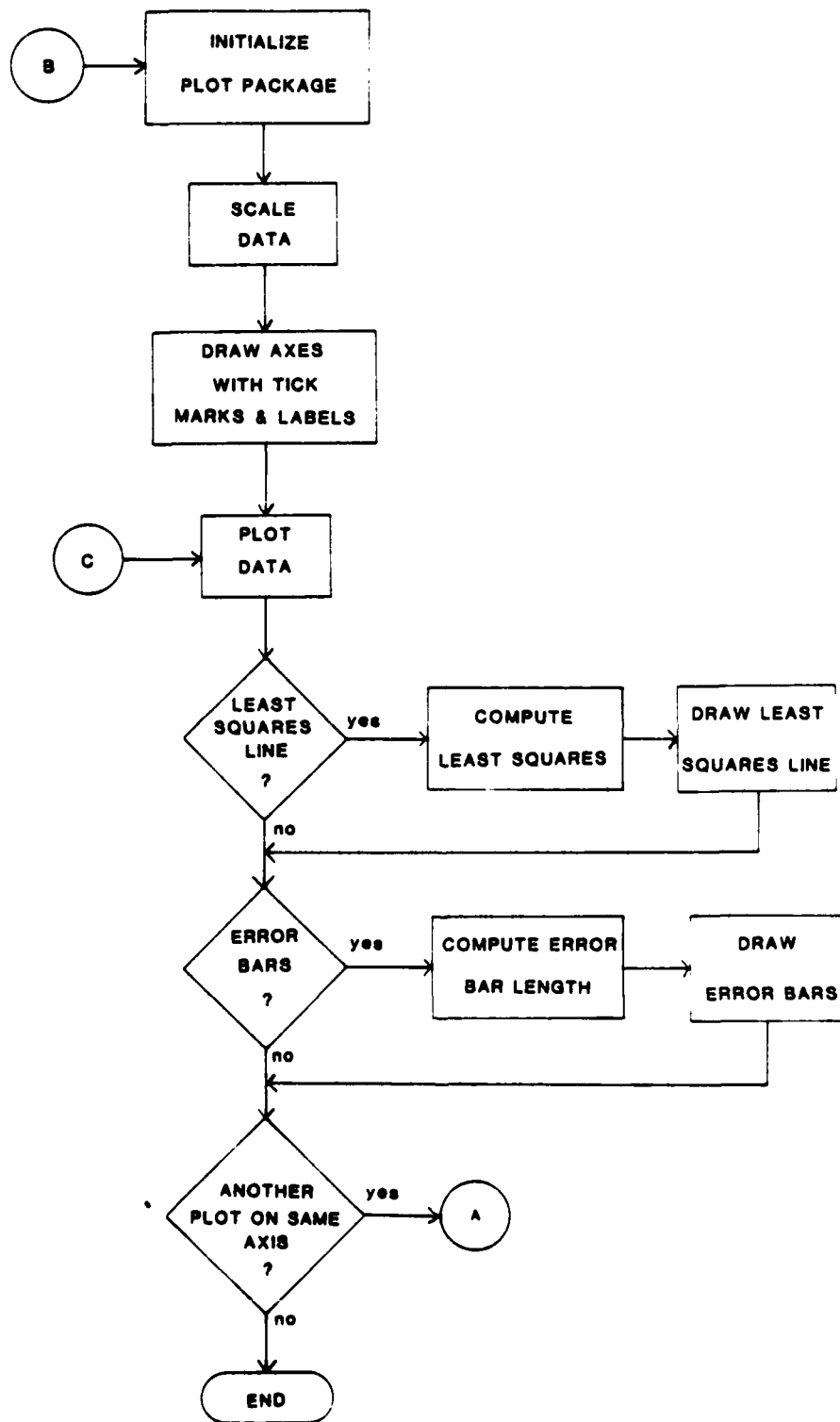
DIF = the interval over which the calculation is computed (FROM-TO+1)

$\Sigma X$  = summation of X values

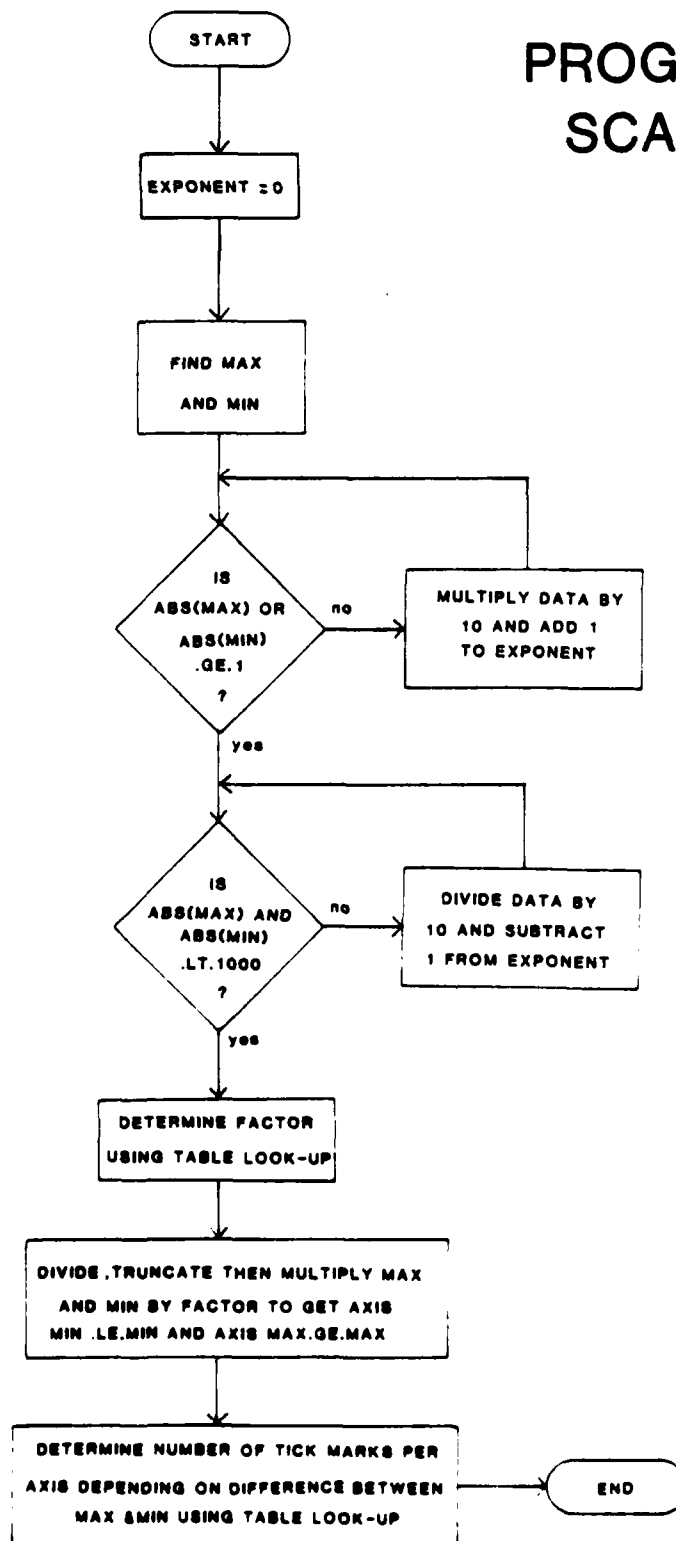
$\Sigma Y$  = summation of Y values

# PROGRAM LPLOT





# PROGRAM SCALE



# PLOTTING EXAMPLES

## 1. LINEAR AXES

### LINE PLOT

#### DATA:

391.51282	.25431687
397.39148	.33236635
403.27008	.40419948
409.14874	.47188836
415.02740	.50006902
420.90607	.50877190
426.78473	.49136621
432.66333	.45517337
438.54199	.40295619
444.42065	.34783810
450.29932	.29423952
456.17798	.24892941
462.05664	.21922916
467.93524	.19685039
473.81390	.18386519
479.69257	.17585301
485.57123	.17474788
491.44989	.17474788
497.32849	.17751071
503.20715	.18414146
509.08582	.19436386
514.96448	.20721093
520.84314	.23456278
526.72180	.26453930
532.60046	.30777734
538.47913	.35640281
544.35767	.41207349
550.23633	.47575635
556.11499	.53598559
561.99365	.59787261
567.87231	.65451026
573.75098	.69691944
579.62964	.72413313
585.50830	.74057186
591.38696	.73394120
597.26550	.71529221
603.14417	.67302108
609.02283	.61983705
614.90149	.56016028
620.78015	.49440533
626.65881	.42243403
632.53748	.35115349
638.41614	.28526038
644.29480	.23097113
650.17346	.18220749
656.05212	.14698163
661.93079	.12170190
667.80933	.97803563E-01
673.68799	.83851367E-01

679.56665	.71694985E-01
685.44531	.65202370E-01
691.32397	.60505591E-01
697.20264	.54427408E-01
703.06130	.51940873E-01
708.95996	.48763640E-01
714.83862	.48763640E-01
720.71716	.47244094E-01

## 2. LINEAR AXES

SYMBOLS WITH CONNECTING LINES PLOT

FOUR DIFFERENT SYMBOL PLOTS ON SAME AXIS

DATA:

1. 2, 10.1  
 4, 10.2  
 6, 10.2  
 8, 10.1  
 10, 10.2  
 12, 10.2  
 14, 10.3  
 16, 10.3  
 18, 10.3  
 20, 10.3  
 22, 10.4  
 24, 10.4  
 26, 10.5  
 28, 10.4  
 30, 10.2  
 32, 10.0  
 34, 9.94  
 36, 10.0  
 38, 9.95  
 40, 10.1

2. 2, 10.0  
 4, 10.1  
 6, 9.94  
 8, 10.0  
 10, 10.1  
 12, 10.1  
 14, 10.3  
 16, 10.2  
 18, 10.0  
 20, 10.3  
 22, 10.3  
 24, 10.4  
 26, 10.3  
 28, 10.3  
 30, 10.1  
 32, 9.91  
 34, 9.82  
 36, 9.92  
 38, 9.88  
 40, 9.94

3. 2, 10.3

4, 10.6  
 6, 10.6  
 8, 10.6  
 10, 10.9  
 12, 10.9  
 14, 11.0  
 16, 11.1  
 18, 11.1  
 20, 11.2  
 22, 11.4  
 24, 11.4  
 26, 11.6  
 28, 11.5  
 30, 11.5  
 32, 11.5  
 34, 11.5  
 36, 11.5  
 38, 11.5  
 40, 11.5

4. 2, 10.2  
 4, 10.2  
 6, 10.3  
 8, 10.3  
 10, 10.4  
 12, 10.4  
 14, 10.4  
 16, 10.5  
 18, 10.4  
 20, 10.5  
 22, 10.7  
 24, 10.7  
 26, 10.8  
 28, 10.6  
 30, 10.3  
 32, 10.3  
 34, 10.2  
 36, 10.2  
 38, 10.1  
 40, 10.3

3. SEMI-LOG AXES (LOGARITHMIC X , LINEAR Y)  
 SYMBOLS WITH CONNECTING LINES PLOT  
 A SINGLE PLOT DIVIDED INTO 3 OVERLAPPING DATA  
 FILES WITH A LEAST SQUARES LINE ON EACH FILE.  
 DATA:

.0016, 2456  
 .0040, 2270  
 .0095, 2033  
 .0155, 1790  
 .0340, 1590  
 .1175, 1535  
 1.24, 1475  
 6.33, 1445  
 32.4, 1245  
 105.3, 1038

445.2,867  
890.4,685  
1480,465

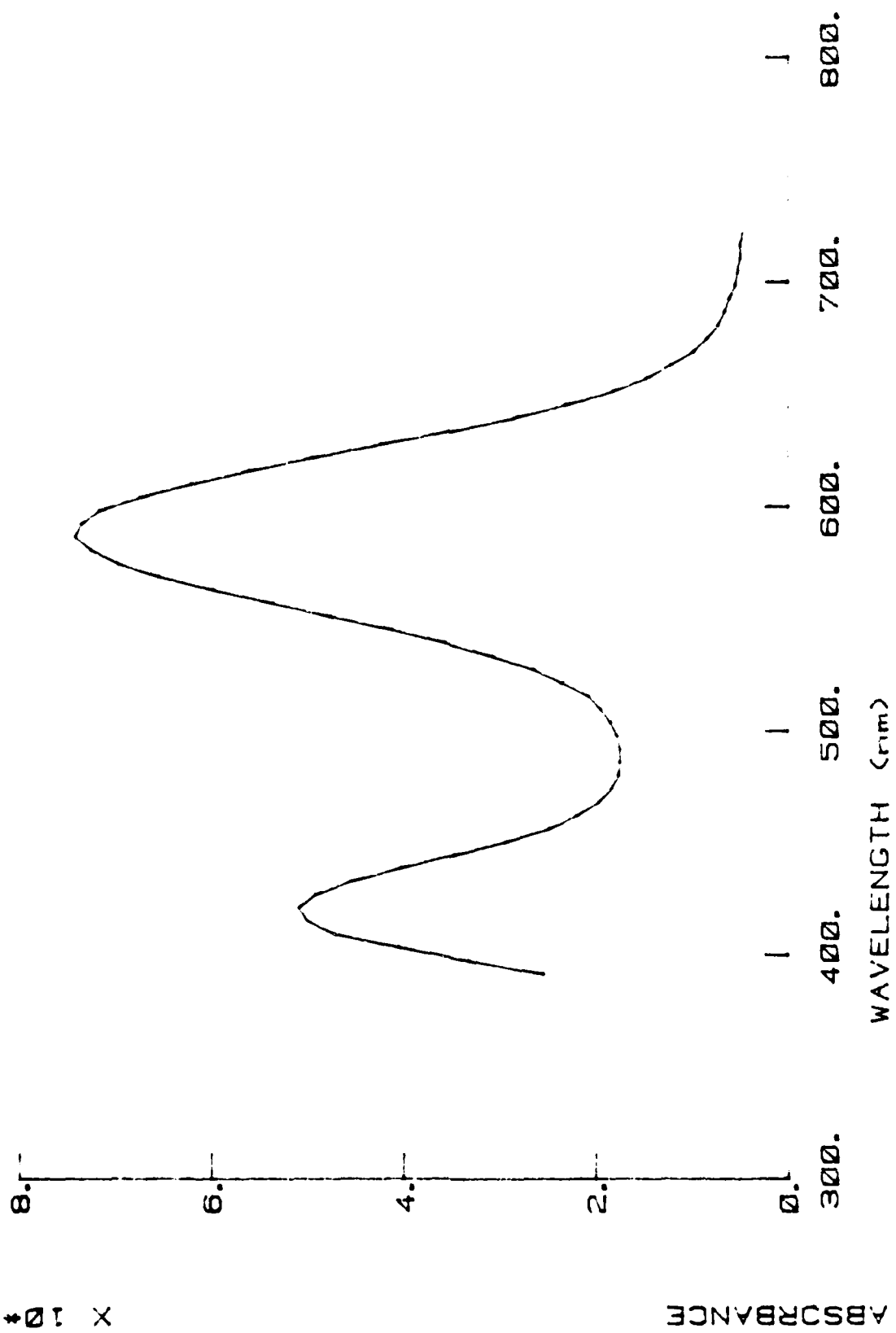
4. LOGARITHMIC AXES  
SYMBOL PLOT WITH X AND Y ERROR BARS  
LEAST SQUARES LINE

DATA:

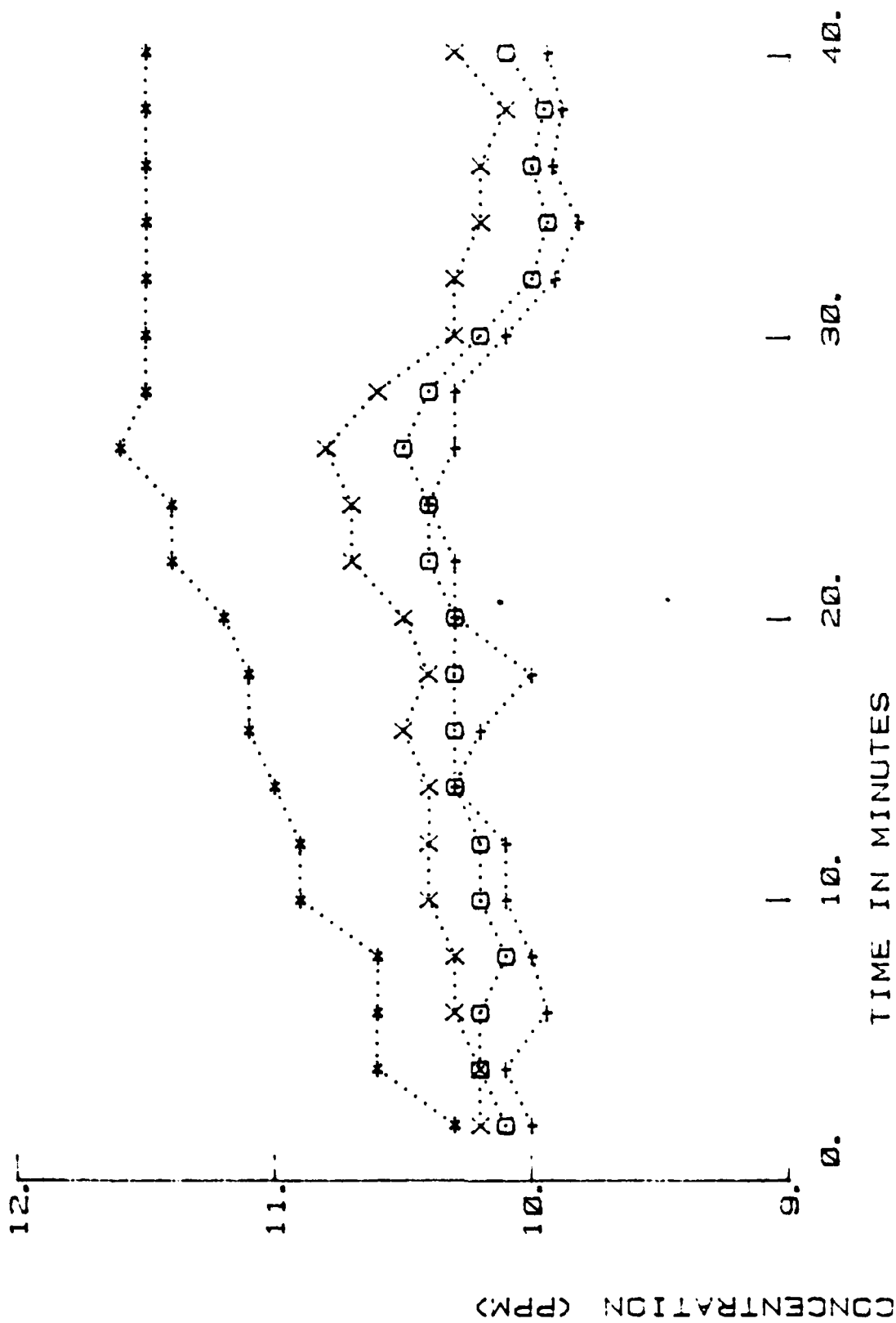
.023,124,.009,48.3  
.79,435,.26,153.5  
8.5,1255,1.54,380  
43.8,10235,7.6,3630  
125,120450,22.4,32540  
840,1350450,165,480000

X 10<sup>4</sup> 1

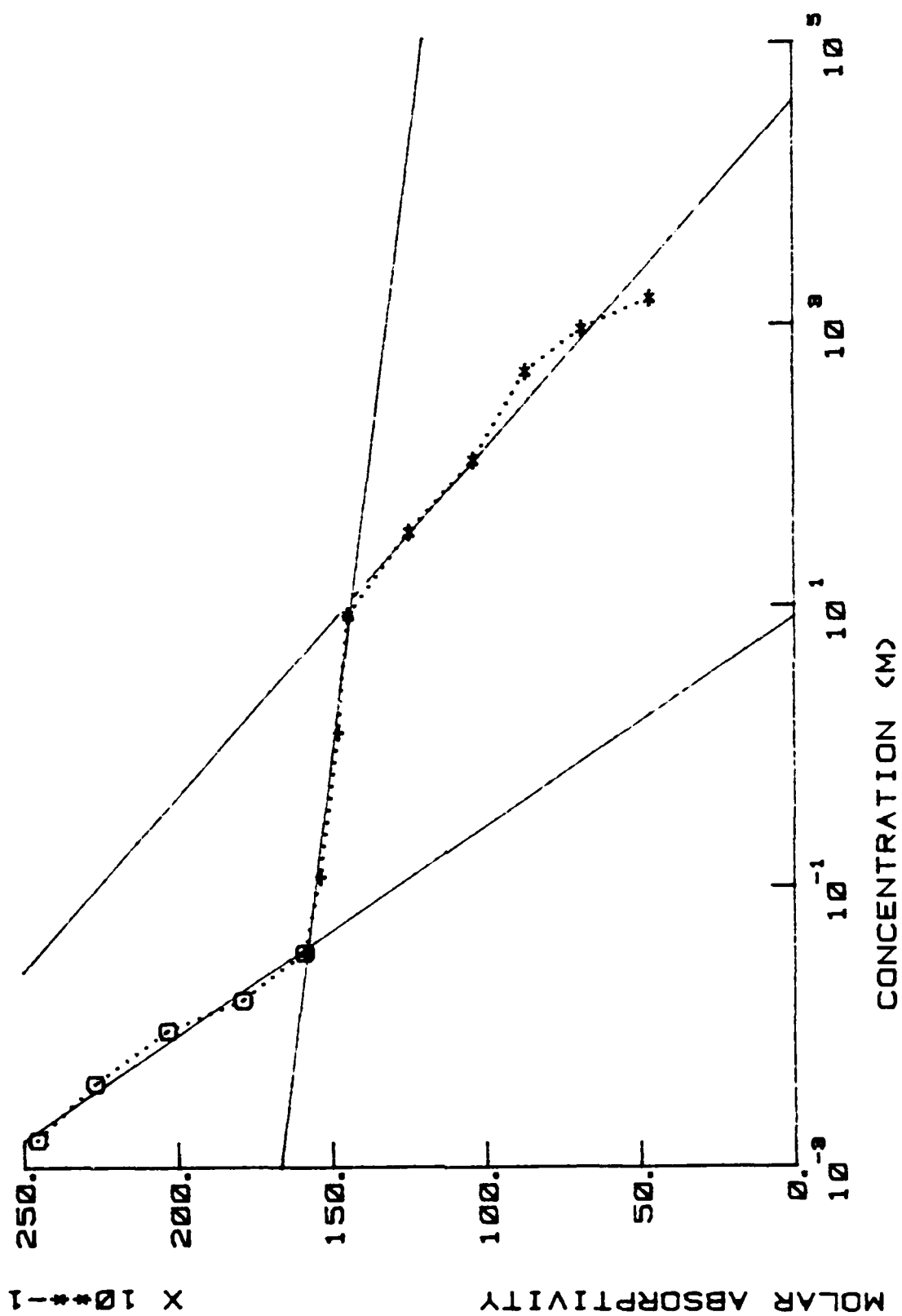
PLOT 1 : UV SPECTRUM



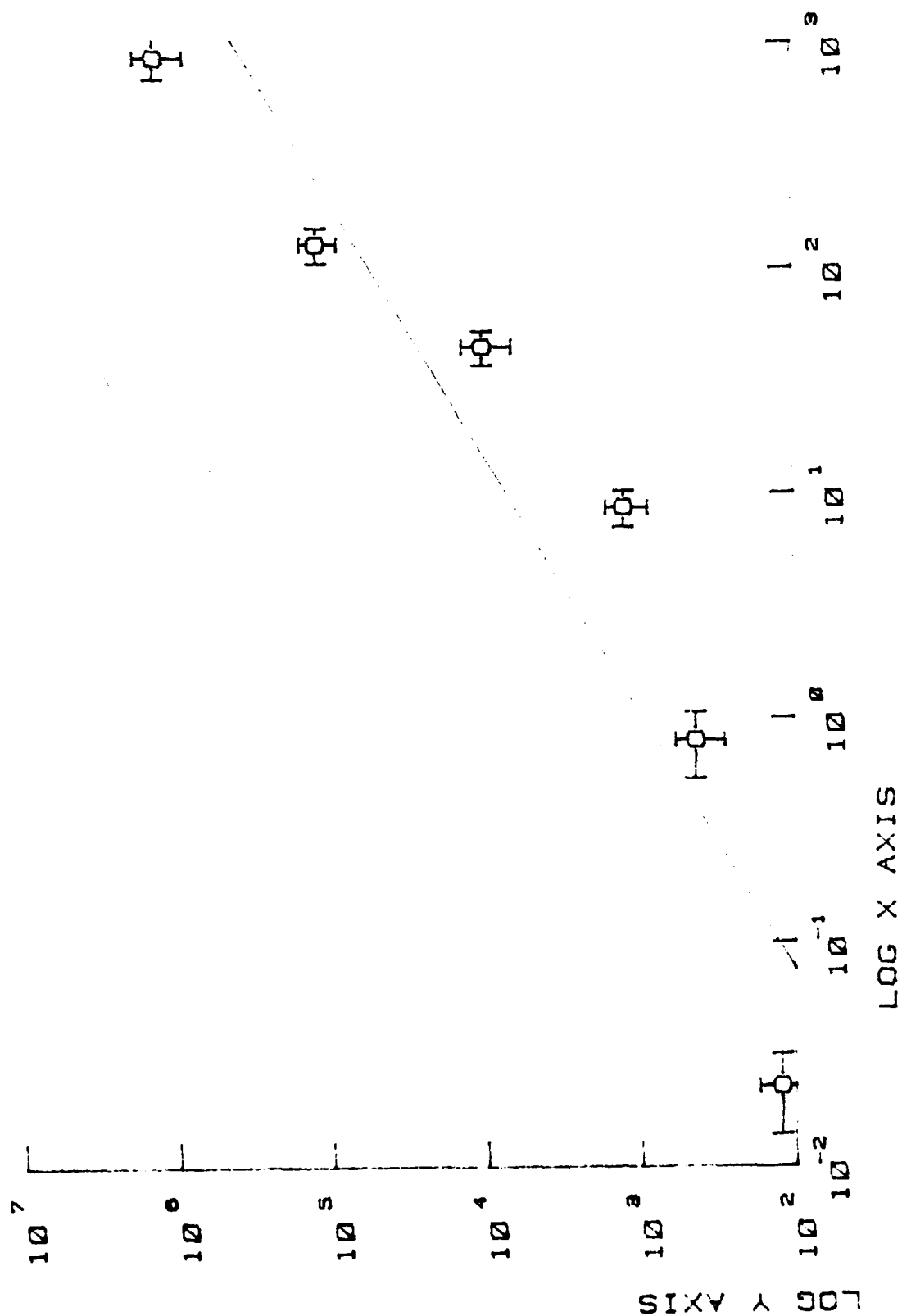
PLOT 2 : O-B . +-P . \*-Cd . X-Fe



PLOT 3: PVP with Fe<sup>3+</sup>



PLOT 4 : EXAMPLE OF A LOG LOG PLOT



## References

1. Hewlett-Packard, RTE Fortran IV Reference Manual, January 1980.
2. Hewlett-Packard, RTE Relocatable Library Reference Manual, December 1978, p. 3-7.
3. Hewlett-Packard, HP 92068A RTE-IVB Terminal User's Reference Manual, January 1980, p. 3-7.
4. Hewlett-Packard, RTE Fortran IV Reference Manual, January 1980, p. 7-4.
5. Hewlett-Packard, HP 92840A Graphics Plotting Software User's Manual, January 1980, p. 5-1.
6. Hewlett-Packard, HP 92068A RTE-IVB Terminal User's Reference Manual, January 1980, p. 1-2, 4-56.

SAMPLE DIALOGUE FOR EXECUTION OF PROGRAM LPLOT

```

:RU,LPLOT
  OUTPUT ON GRAPHICS TERMINAL TYPE 0
      ON PLOTTER TYPE 1
0
  *** SOURCE OF DATA ***
  DISC FILE TYPE 0
  CARTRIDGE TAPE TYPE 1
  TERMINAL TYPE 2
0
  ENTER NAME OF DATA FILE
PVP1
  *** TYPE OF PLOT ***
  LINE PLOT TYPE 1
  SYMBOLS CONNECTED WITH LINES TYPE 2
  SYMBOLS PLOT TYPE 3
2
  *** TYPE OF AXES ***
  LINEAR X AND Y TYPE 0
  LOGARITHMIC X AND Y TYPE 1
  LOGARITHMIC X AND LINEAR Y TYPE 2
  LOGARITHMIC Y AND LINEAR X TYPE 3
3
  ENTER X-LABEL,Y-LABEL & TITLE ON 3 SEPARATE LINES
TIME-MINUTES
LOG-ABSORBANCE
PVP WITH FE+3
  *** ORIGIN LOCATION ***
  TO FORCE LINEAR AXIS TO START AT 0 TYPE 0
  TO LET SCALE DETERMINE ORIGIN TYPE 1
1
  *** SCALING ***
  AUTOMATIC TYPE 0
  MANUAL TYPE 1
0
  LEAST SQUARES LINE ?
                                NO TYPE 0
                                YES TYPE 1
1
  ERROR BARS ?
                                NONE TYPE 0
                                X TYPE 1
                                Y TYPE 2
                                X & Y TYPE 3
0
  TO EXIT TYPE 0
  IF YOU WANT ANOTHER PLOT ON SAME AXES
  AND ALL X & Y VALUES ARE WITHIN THE
  SCALES OF THE FIRST PLOT TYPE 1
0
:
```

PAGE 0001 FTN. 9:10 AM MON., 26 JAN., 1981

```
0001 FTN4,L
0002 PROGRAM LPL0T(4,99)
0003 C LOGARITHMIC & LINEAR PLOTTING ROUTINE FOR THE HP 9872B PLOTTER
0004 C MAXIMUM 256 PTS. ALLOWED, X AND Y MAY BE INPUT FROM A DISC
0005 C FILE, CARTRIDGE TAPE OR KEYBOARD FORMATTED IN (X,Y) PAIRS
0006 C FOR EXPERIMENTAL ERROR BARS FORMAT IS (X,Y,DELTAX,DELTAY).
0007 C
0008 COMMON IGCB(192)
0009 DIMENSION IDCB(144),NAME(3),IAX(15),IAY(15),IHED(20),IBUF(40)
0010 DIMENSION X(256),Y(256),DELX(256),DELY(256)
0011 DIMENSION IPRAM(5),IOBUF(20),JCHAR(6)
0012 DATA JCHAR/1H0,1H+,1H*,1HX,1H#,1H$/
0013 CALL RMPAR(IPRAM)
0014 LU=IPRAM(1)
0015 LUG=IPRAM(2)
0016 C
0017 C SET DEFAULT VALUES
0018 C
0019 ILUG=0
0020 IFMT=0
0021 JJ=1
0022 IAXES=0
0023 IZERO=1
0024 ISCAL=0
0025 ILINE=C
0026 IBAR=0
0027 IDONE=0
0028 IPEN=1
0029 LERR=0
0030 C
0031 C ESTABLISH OUTPUT DEVICE
0032 C
0033 IF(LUG.NE.0)GOTO 15
0034 WRITE(LU,10)
0035 10 FORMAT(" OUTPUT ON GRAPHICS TERMINAL TYPE 0",/,
0036 1 " ON PLOTTER TYPE 1")
0037 READ(LU,*)ILUG
0038 C DEFINE LU AND ID NUMBERS OF PLOTTER AND GRAPHICS TERMINAL
0039 LUG=24
0040 IF(ILUG.EQ.1)LUG=20
0041 15 KK=0
0042 IF(LU.LE.0)LU=1
0043 ID=1
0044 IF(LUG.EQ.20)ID=2
0045 C TO CENTER CHARACTER SET HALF WIDTH & HALF HEIGHT
0046 C DEPENDING ON IF USING PLOTTER OR CRT
0047 HW=0.5
0048 HH=0.5
0049 IF(ID.EQ.1) GOTO 20
0050 HW=0.333
0051 HH=0.25
0052 20 KK=KK+1
0053 C
0054 C DETERMINE DATA SOURCE AND READ DATA
0055 C
```

```

PAGE 0002 L PLOT 9:10 AM MON., 26 JAN., 1981
0056 25 WRITE(LU,30)
0057 30 FORMAT(" *** SOURCE OF DATA ***"/
0058 1 " DISC FILE TYPE 0"/
0059 2 " CARTRIDGE TAPE TYPE 1"/
0060 3 " TERMINAL TYPE 2")
0061 READ (LU,*) IFMT
0062 IF(IFMT.EQ.0) GOTO 70
0063 35 IF(IFMT.EQ.1)WRITE(LU,40)
0064 IF(IFMT.NE.1)WRITE(LU,45)
0065 40 FORMAT(" ENTER LU NUMBER OF CTU")
0066 45 FORMAT(" ENTER LU NUMBER OF TERMINAL")
0067 READ(LU,*)LUT
0068 IF(LUT.GT.30) GOTO 35
0069 50 WRITE(LU,55)
0070 55 FORMAT(" ENTER NUMBER OF DATA POINTS (MAX=256)")
0071 READ(LU,*)NP
0072 IF(NP.GT.256) GOTO 50
0073 IF(IFMT.NE.1)WRITE(LU,60)NP
0074 60 FORMAT(" ENTER ",I3," DATA POINTS")
0075 DO65 I=1,NP
0076 65 READ (LUT,*) X(I),Y(I),DELX(I),DELY(I)
0077 GOTO 110
0078 70 WRITE(LU,75)
0079 75 FORMAT(" ENTER NAME OF DATA FILE")
0080 READ(LU,80)NAME
0081 80 FORMAT(3A2)
0082 CALL OPEN(IDCB,IERR,NAME,3)
0083 IF(IERR.LT.0)WRITE(LU,85)NAME
0084 IF(IERR.LT.0)GOTO 25
0085 85 FORMAT(" FILE ",3A2," DOESN'T EXIST OR IS ALREADY OPEN")
0086 K=1
0087 90 DO95 J=1,40
0088 95 IBUF(J)=0
0089 CALL READF(IDCB,IERR,IBUF,40,LEN)
0090 IF(LEN.EQ.-1) GOTO 100
0091 IF(IERR.LT.0)WRITE(LU,105)IERR
0092 CALL CODE
0093 READ(IBUF,*)X(K),Y(K),DELX(K),DELY(K)
0094 K=K+1
0095 GO TO 90
0096 100 NP=K-1
0097 CALL CLOSE(IDCB,IERR)
0098 IF(IERR.LT.0)WRITE(LU,105)IERR
0099 105 FORMAT(" DISC I/O ERROR = "I4)
0100 C
0101 C SCALE DATA TO FIRST PLOT (FOR MULTIPLE PLOTTING ONLY)
0102 C
0103 110 IF(KK.EQ.1) GOTO 165
0104 IF(IAXES.EQ.1)GOTO 145
0105 IF(IEXP.EQ.0.AND.JEXP.EQ.0)GOTO 145
0106 DO140 K=1,NP
0107 IF(IEXP)115,125,120
0108 115 X(K)=X(K)/(10.**IABS(IEXP))
0109 GOTO 125
0110 120 X(K)=X(K)*(10.**IABS(IEXP))

```

```

PAGE 0003 LPLLOT 9:10 AM MON.. 26 JAN., 1931
0111 125 IF(JEXP)130,140,135
0112 130 Y(K)=Y(K)/(10.**IABS(JEXP))
0113 140 GOTO 140
0114 135 Y(K)=Y(K)*(10.**IABS(JEXP))
0115 140 CONTINUE
0116 145 IF(IAXES.NE.1.AND.IAXES.NE.2)GOTO 155
0117 150 DO150 K=1,NP
0118 X(K)=ALOGT(X(K))
0119 IF(DEIX(K).EQ.0.)GOTO 150
0120 DELX(K)=ALOGT(ABS(DEIX(K)))
0121 150 CONTINUE
0122 155 IF(IAXES.NE.1.AND.IAXES.NE.3)GOTO 165
0123 160 DO160 K=1,NP
0124 Y(K)=ALOGT(Y(K))
0125 IF(DELY(K).EQ.0.)GOTO 160
0126 DELY(K)=ALOGT(ABS(DELY(K)))
0127 160 CONTINUE
0128 C
0129 C DETERMINE PLOTTING FEATURES
0130 C
0131 165 WRITE(LU,170)
0132 170 FORMAT(" *** TYPE OF PLOT **",/
0133 1 " LINE PLOT TYPE 1",/
0134 2 " SYMBOLS CONNECTED WITH LINES TYPE 2",/
0135 3 " SYMBOLS PLOT TYPE 3",)
0136 READ(LU,*)JJ
0137 IF(KK.GT.1)GOTO 300
0138 WRITE(LU,175)
0139 175 FORMAT(" *** TYPE OF AXES **",/
0140 1 " LINEAR X AND Y TYPE 0",/
0141 2 " LOGARITHMIC X AND Y TYPE 1",/
0142 3 " LOGARITHMIC X AND LINEAR Y TYPE 2",/
0143 4 " LOGARITHMIC Y AND LINEAR X TYPE 3",)
0144 READ(LU,*)IAXES
0145 WRITE(LU,180)
0146 180 FORMAT(" ENTER X-LABEL,Y-LABEL & TITLE ON 3 SEPAFATE LINES")
0147 READ(LU,185) IAX,IAY,IHED
0148 185 FORMAT(15A2/,15A2/,20A2)
0149 IF(IAXES.EQ.1)GOTO 195
0150 WRITE(LU,190)
0151 190 FORMAT(" *** ORIGIN LOCATION **",/
0152 1 " TO FORCE LINEAR AXIS TO START AT 0 TYPE 0",/
0153 2 " TO LET SCALE DETERMINE ORIGIN TYPE 1")
0154 READ(LU,*)IZERO
0155 195 WRITE(LU,200)
0156 200 FORMAT(" *** SCALING **",/
0157 1 " AUTOMATIC TYPE 0",/
0158 2 " MANUAL TYPE 1")
0159 READ(LU,*)ISCAL
0160 IF(ISCAL.EQ.1)GOTO 210
0161 WRITE(LU,205)
0162 205 FORMAT(" ENTER XMIN,XMAX,YMIN,YMAX")
0163 READ(LU,*)XMIN,XMAX,YMIN,YMAX
0164 C
0165 C INITIALIZE PLOTTER

```

```

PAGE 0004 LPL0T 9:10 AM MON., 26 JAN., 1981
0166 C
0167 210 CALL PLOT(IGCB,ID,4,LUG,IOBUF,20)
0168 CALL SETAP(IGCB,2.0)
0169 CALL VIEWP(IGCB,0.,135.,0.,100.)
0170 CALL WINDW(IGCB,0.,150.,0.,100.)
0171 CALL CSIZE(IGCB,3.)
0172 CALL FXD(IGCB,0)
0173 CALL PEN(IGCB,1)
0174 C
0175 C SCALE DATA
0176 C
0177 IEXP=0
0178 JEXP=0
0179 IF(IAXES.NE.0.AND.IAXES.NE.3)GOTO 215
0180 CALL SCALE (X,NP,XMIN,XMAX,XTIC,IZERO,IEXP,LU,ISCAL)
0181 215 IF(IAXES.NE.0.AND.IAXES.NE.2)GOTO 220
0182 CALL SCALE (Y,NP,YMIN,YMAX,YTIC,IZERO,JEXP,LU,ISCAL)
0183 220 IF(IAXES.NE.1.AND.IAXES.NE.2)GOTO 230
0184 CALL LSCAL(X,NP,XMIN,XMAX,XTIC,LU,ISCAL,LERR)
0185 IF(LERR.EQ.1)GOTO 490
0186 DO225 K=1,NP
0187 X(K)=ALOGT(X(K))
0188 IF(DEIX(K).EQ.0.)GOTO 225
0189 DELX(K)=ALOGT(ABS(DEIX(K)))
0190 225 CONTINUE
0191 230 IF(IAXES.NE.1.AND.IAXES.NE.3)GOTO 240
0192 CALL LSCAL(Y,NP,YMIN,YMAX,YTIC,LU,ISCAL,LERR)
0193 IF(LERR.EQ.1)GOTO 490
0194 DO235 K=1,NP
0195 Y(K)=ALOGT(Y(K))
0196 IF(DELY(K).EQ.0.)GOTO 235
0197 DELY(K)=ALOGT(ABS(DELY(K)))
0198 235 CONTINUE
0199 C
0200 C WRITE AXES LABELS AND TITLE
0201 C
0202 240 CALL MOVE(IGCB,35.,1.)
0203 IF(IEXP.NE.0) GOTO 250
0204 CALL LABEL(IGCB)
0205 WRITE(LUG,245)IAX
0206 245 FORMAT(15A2)
0207 GOTO 260
0208 250 CALL LABEL(IGCB)
0209 WRITE(LUG,255)IAX,IEXP
0210 255 FORMAT(15A2," X 10**",I2)
0211 260 CALL MOVE(IGCB,3.,12.)
0212 CALL LDIR(IGCB,+1.57)
0213 IF(JEXP.NE.0) GOTO 265
0214 CALL LABEL(IGCB)
0215 WRITE(LUG,245)IAY
0216 GOTO 270
0217 265 CALL LABEL(IGCB)
0218 WRITE(LUG,255)IAY,JEXP
0219 270 CALL MOVE(IGCB,40.,90.)
0220 CALL LDIR(IGCB,0.)

```

```

PAGE 0005 L PLOT 9:10 AM MON., 26 JAN., 1981
0221 CALL LABEL(IGCB)
0222 WRITE(LUG,275)IHED
0223 275 FORMAT(20A2)
0224 C
0225 C DRAW AXES
0226 C
0227 IF(IAXES.NE.1.AND.IAXES.NE.2)GOTO 280
0228 CALL LOGAX(1,XMIN,XMAX,XTIC,LUG)
0229 280 IF(IAXES.NE.1.AND.IAXES.NE.3)GOTO 285
0230 CALL LOGAX(2,YMIN,YMAX,YTIC,LUG)
0231 285 IF(IAXES.NE.0.AND.IAXES.NE.3)GOTO 290
0232 CALL LINAX(1,XMIN,XMAX,XTIC,LUG)
0233 290 IF(IAXES.NE.0.AND.IAXES.NE.2)GOTO 295
0234 CALL LINAX(2,YMIN,YMAX,YTIC,LUG)
0235 295 CALL VIEWP(IGCB,17.,120.,10.,80.)
0236 CALL WINDOW(IGCB,XMIN,XMAX,YMIN,YMAX)
0237 C
0238 C PLOT DATA POINTS
0239 C
0240 300 MN=0
0241 IF(JJ.EQ.2)M4=1
0242 CALL LINE(IGCB,MN)
0243 DO 310 K=1,NP
0244 IF(K.EQ.1.OR.JJ.EQ.3) GOTO 305
0245 CALL DRAW(IGCB,X(K),Y(K))
0246 C MOVE TO (X,Y),CENTER CHAR.,PLOT CHAR.,MOVE "CP" BACK TO (X,Y)
0247 305 CALL MOVE(IGCB,X(K),Y(K))
0248 IF(JJ.EQ.1) GOTO 310
0249 CALL CPLOT(IGCB,-HW,-HH,-2)
0250 CALL LABEL(IGCB)
0251 WRITE(LUG,315)JCHAR(KK)
0252 CALL MOVE(IGCB,X(K),Y(K))
0253 310 CONTINUE
0254 315 FORMAT(1A1)
0255 CALL PENUP(IGCB)
0256 CALL LINE(IGCB,0)
0257 WRITE(LU,320)
0258 320 FORMAT(" LEAST SQUARES LINE ?",/,22X,"NO TYPE 0",/,
0259 1 22X,"YES TYPE 1")
0260 READ(LU,*) ILINE
0261 WRITE(LU,325)
0262 325 FORMAT(" ERROR BARS ?",/,
0263 1 " NONE TYPE 0",/,
0264 2 " X TYPE 1",/,
0265 3 " Y TYPE 2",/,
0266 4 " X & Y TYPE 3")
0267 READ(LU,*)IBAR
0268 IF(ILINE.EQ.0) GOTO 360
0269 C
0270 C CALCULATE START AND END POINTS OF LEAST SQUARES LINE
0271 C
0272 CALL LSREG(X,Y,1,NP,SLOPE,YINT,S1,S2)
0273 XST=XMIN
0274 YST=YINT
0275 YST=(SLOPE*XMIN)+YINT

```

```

PAGE 0000 L PLOT 9:10 AM MON., 23 JAN., 1981
0276 330 IF(YINT.LE.YMAX) GO TO 335
0277 XST=(YMAX-YINT)/SLOPE
0278 YST=YMAX
0279 335 IF(YINT.GE.YMIN) GO TO 340
0280 XST=(YMIN-YINT)/SLOPE
0281 YST=YMIN
0282 340 XEND=XMAX
0283 YEND=(SLOPE*XMAX)+YINT
0284 IF(YEND.LE.YMAX) GO TO 345
0285 XEND=(YMAX-YINT)/SLOPE
0286 YEND=YMAX
0287 345 IF(YEND.GE.YMIN) GO TO 350
0288 YEND=YMIN
0289 XEND=(YMIN-YINT)/SLOPE
0290 C
0291 C PLOT LEAST SQUARES LINE
0292 C
0293 350 IF(ILINE.EQ.0) GOTO 360
0294 CALL MOVE(IGCB,XST,YST)
0295 CALL DRAW(IGCB,XEND,YEND)
0296 CALL PENUP(IGCB)
0297 IF(IAXES.NE.0) GOTO 360
0298 LUX=LU
0299 IF(ID.EQ.2) LUX=6
0300 SLOPE=SLOPE*(10.**((JEXP-IEXP)*(-1)))
0301 YINT=YINT*(10.**((JEXP)*(-1)))
0302 S1=S1*(10.**((JEXP-IEXP)*(-1)))
0303 S2=S2*(10.**((JEXP)*(-1)))
0304 WRITE(LUX,355)SLOPE,S1,YINT,S2
0305 355 FORMAT(" SLOPE="G12.5", +-"G12.6/" Y-int="G12.5", +-"G12.6)
0306 C
0307 C COMPUTE AND PLOT ERROR BARS
0308 C
0309 360 IF(IBAR.EQ.0) GOTO 465
0310 C SCALE DEVIATIONS BY VALUE OF EXPONENT
0311 DO390 K=1,NP
0312 IF(IEXP)365,375,370
0313 365 DELX(K)=ABS(DELX(K)/(10.**IABS(IEXP)))
0314 GOTO 375
0315 370 DELX(K)=ABS(DELX(K)*(10.**IABS(IEXP)))
0316 375 IF(JEXP)380,390,385
0317 380 DELY(K)=ABS(DELY(K)/(10.**IABS(JEXP)))
0318 GOTO 390
0319 385 DELY(K)=ABS(DELY(K)*(10.**IABS(JEXP)))
0320 390 CONTINUE
0321 NOBAR=0
0322 C DON'T DRAW ERROR BARS LESS THAN 1.05% X OR 1.25% Y OF LINEAR AXIS
0323 BARX=(XMAX-XMIN)*0.0105
0324 BARY=(YMAX-YMIN)*0.0125
0325 DO455 K=1,NP
0326 IF(IAXES.NE.1.AND.IAXES.NE.2)GOTO 395
0327 C DON'T DRAW ERROR BARS LESS THAN 13.5% X OR 18% Y OF LOG POINT
0328 BARX=(ABS(10.**X(K)))*0.135
0329 XBAR1=ALOGT(ABS((10.**X(K))- (10.**DELX(K))))
0330 XBAR2=ALOGT(ABS((10.**X(K))+(10.**DELX(K))))

```

```

PAGE 0007 L PLOT 9:10 AM MON., 26 JAN., 1981
0331 395 IF(IAXES.NE.1.AND.IAXES.NE.3)GOTO 400
0332 BARY=(ABS(10.**Y(K)))*0.18
0333 YBAR1=ALOGT(ABS((10.**Y(K))+(10.**DELY(K))))
0334 YBAR2=ALOGT(ABS((10.**Y(K))-(10.**DELY(K))))
0335 400 IF(IAXES.NE.0.AND.IAXES.NE.3)GOTO 405
0336 XBAR1=X(K)-DELX(K)
0337 XBAR2=X(K)+DELX(K)
0338 405 IF(IAXES.NE.0.AND.IAXES.NE.2)GOTO 410
0339 YBAR1=Y(K)+DELY(K)
0340 YBAR2=Y(K)-DELY(K)
0341 410 GOTO(420,440,420) IBAR
0342 415 GOTO 465
0343 C DRAW X ERROR BARS
0344 420 IF(IAXES.NE.1.AND.IAXES.NE.2)GOTO 425
0345 IF(10.**DELX(K).LT.BARX)NOBAR=1
0346 IF(10.**DELX(K).LT.BARX)GOTO 435
0347 GOTO 430
0348 425 IF(DELX(K).LT.BARX)NOBAR=1
0349 IF(DELX(K).LT.BARX)GOTO 435
0350 430 CALL MOVE(IGCB,X(K),Y(K))
0351 CALL CPlot(IGCB,-0.5,0.,-2)
0352 CALL DRAW(IGCB,XBAR1,Y(K))
0353 CALL CPlot(IGCB,0.,-0.3,-2)
0354 CALL CPlot(IGCB,0.,0.6,-1)
0355 CALL MOVE(IGCB,X(K),Y(K))
0356 CALL CPlot(IGCB,0.5,0.,-2)
0357 CALL DRAW(IGCB,XBAR2,Y(K))
0358 CALL CPlot(IGCB,0.,-0.3,-2)
0359 CALL CPlot(IGCB,0.,0.6,-1)
0360 435 IF(IBAR.EQ.1) GOTO 455
0361 C DRAW Y ERROR BARS
0362 440 IF(IAXES.NE.1.AND.IAXES.NE.3)GOTO 445
0363 IF(10.**DELY(K).LT.BARY)NOBAR=1
0364 IF(10.**DELY(K).LT.BARY)GOTO 455
0365 GOTO 450
0366 445 IF(DELY(K).LT.BARY) NOBAR=1
0367 IF(DELY(K).LT.BARY) GOTO 455
0368 450 CALL MOVE(IGCB,X(K),Y(K))
0369 CALL CPlot(IGCB,0.,0.3,-2)
0370 CALL DRAW(IGCB,X(K),YBAR1)
0371 CALL CPlot(IGCB,-0.3,0.,-2)
0372 CALL CPlot(IGCB,0.6,0.,-1)
0373 CALL MOVE(IGCB,X(K),Y(K))
0374 CALL CPlot(IGCB,0.,-0.3,-2)
0375 CALL DRAW(IGCB,X(K),YBAR2)
0376 CALL CPlot(IGCB,-0.3,0.,-2)
0377 CALL CPlot(IGCB,0.6,0.,-1)
0378 455 CONTINUE
0379 CALL PENUP(IGCB)
0380 IF(NOBAR.EQ.1) WRITE(LU,460)
0381 460 FORMAT(" *** SOME ERROR BARS WERE TOO SMALL TO PLOT ***")
0382 465 WRITE(LU,470)
0383 470 FORMAT(" TO EXIT TYPE 0",/,
0384 1 " IF YOU WANT ANOTHER PLOT ON SAME AXES",/,
0385 2 " AND ALL X & Y VALUES ARE WITHIN THE",/,

```

```

PAGE 0008 LPLCT 9:10 AM MON., 26 JAN., 1981
0386      3      " SCALES OF THE FIRST PLOT          TYPE 1")
0387      READ(LU,*)IDONE
0388      IF(KK.EQ.6) GOTO 490
0389  C      SELECT PEN COLOR IF USING PLOTTER
0390      IF(ID.NE.2.OR.IDONE.NE.1) GOTO 485
0391  475     WRITE(LU,480)
0392  480     FORMAT(" *** PEN COLOR SELECTION ***",/,
0393      1      "      BLACK          TYPE 1",/,
0394      2      "      RED            TYPE 2",/,
0395      3      "      GREEN          TYPE 3",/,
0396      4      "      BLUE          TYPE 4")
0397      READ(LU,*) IPEN
0398      IF(IPEN.LT.1.OR.IPEN.GT.4) GOTO 475
0399      CALL PEN(IGCB,IPEN)
0400  485     IF(IDONE.EQ.1) GOTO 20
0401  490     CALL PEN(IGCB,0)
0402      CALL PLOT(IGCB,ID,0)
0403      END
FTN4 COMPILER: HP92060-15092 REV. 2001 (791101)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 05735      COMMON = 00192

```

```

0001 FIN4,L
0002 SUBROUTINE SCALE (A,NP,AMIN,AMAX,TIC,IZERO,IEXP,LU,ISCAL)
0003 C THIS ROUTINE COMPUTES MAX & MIN VALUES, SCALES DATA TO
0004 C E FORMAT AND DETERMINES THE NUMBER OF TICK MARKS PER AXIS
0005 C A - ARRAY TO BE SCALED
0006 C NP - NUMBER OF POINTS IN ARRAY A
0007 C AMIN - MINIMUM VALUE OF A
0008 C AMAX - MAXIMUM VALUE OF A
0009 C TIC - NUMBER OF TICK MARKS ON AXIS
0010 C IZERO - SET TO 0 TO FORCE ORIGIN TO ZERO, NORMALLY = 1
0011 C IEXP - EXPONENT OF BASE 10 TO WHICH A IS RAISED
0012 C LU - LOGICAL UNIT NUMBER OF TERMINAL
0013 C ISCAL - SCALING, 0 FOR AUTOMATIC, 1 FOR MANUAL
0014 DIMENSION A(256)
0015 ISML=0
0016 C FIND MAX AND MIN VALUES OF A
0017 IF(ISCAL.EQ.1)GOTO 25
0018 AMAX=A(1)
0019 AMIN=A(1)
0020 DO10 I=1,NP
0021 IF(A(I).GT.AMAX)AMAX=A(I)
0022 IF(A(I).LT.AMIN)AMIN=A(I)
0023 10 CONTINUE
0024 IF(AMAX.NE.AMIN)GOTO 15
0025 AMAX=AMAX+0.01
0026 AMIN=AMIN-0.01
0027 15 IF(IZERO.NE.0.OR.AMIN.GE.0) GOTO 25
0028 WRITE(LU,20)
0029 20 FORMAT(" ** DATA HAS NEGATIVE VALUES ORIGIN CAN'T BE ZERO **"/)
0030 IZERO=1
0031 25 IF(IZERO.EQ.0)AMIN=0.
0032 IEXP=0
0033 DIF=ABS(AMAX-AMIN)
0034 C IF DIF IS LESS THAN 1/2 MAGNIFY SCALE
0035 IF(DIF.GT.0.5)GOTO 40
0036 ISML=1
0037 30 DO35 K=1,NP
0038 35 A(K)=A(K)*10.
0039 AMIN=AMIN*10.
0040 AMAX=AMAX*10.
0041 IEXP=IEXP+1
0042 DIF=DIF*10.
0043 IF(DIF.LT.1.)GOTO 30
0044 GOTO 60
0045 C SCALE DATA TO E FORMAT AND SAVE EXPONENT (IEXP)
0046 40 IF(ABS(AMAX).GE.1.OR.ABS(AMIN).GE.1) GOTO 50
0047 DO45 K=1,NP
0048 45 A(K)=A(K)*10.
0049 AMIN=AMIN*10.
0050 AMAX=AMAX*10.
0051 IEXP=IEXP+1
0052 GOTO 40
0053 50 IF(ABS(AMAX).LE.1000.AND.ABS(AMIN).LE.1000) GOTO 60
0054 DO55 K=1,NP
0055 55 A(K)=A(K)/10.

```

PAGE 0002 SCALE 9:26 AM WED., 14 JAN.. 1981

```
0056 AMIN=AMIN/10.
0057 AMAX=AMAX/10.
0058 IEXP=IEXP-1
0059 GOTO 50
0060 C DETERMINE INTERVAL FACTOR
0061 50 DIF=ABS(AMAX-AMIN)
0062 IF(DIF.GT.5.)GOTO 65
0063 FACTR=1.0
0064 GOTO 100
0065 65 IF(DIF.GT.10.)GOTO 70
0066 FACTR=2.0
0067 GOTO 100
0068 70 IF(DIF.GT.25.)GOTO 75
0069 FACTR=5.0
0070 GOTO 100
0071 75 IF(DIF.GT.50)GOTO 80
0072 FACTR=10.0
0073 GOTO 100
0074 80 IF(DIF.GT.125) GOTO 85
0075 FACTR=25.
0076 GOTO 100
0077 85 IF(DIF.GT.250)GOTO 90
0078 FACTR=50.
0079 GOTO 100
0080 90 IF(DIF.GT.500)GOTO 95
0081 FACTR=100.
0082 GOTO 100
0083 95 FACTR=200.
0084 C HANDLE ROUND-OFF ERROR
0085 100 ONE=0.99999
0086 IF(ISML.EQ.1)ONE=0.0
0087 IF(AMIN.GE.0.)GOTO 105
0088 C **** FOR NEGATIVE NUMBERS ****
0089 TRUNK=(AMIN/FACTR)-ONE
0090 AMIN=AMIN-(TRUNK)*FACTR
0091 IF(AMAX.GE.0) GOTO 110
0092 TRUNK=AMAX/FACTR
0093 AMAX=AMAX-(TRUNK)*FACTR
0094 GOTO 115
0095 C **** FOR POSITIVE NUMBERS ****
0096 105 TRUNK=AMIN/FACTR
0097 AMIN=AMIN-(TRUNK)*FACTR
0098 110 TRUNK=(AMAX/FACTR)+ONE
0099 AMAX=AMAX-(TRUNK)*FACTR
0100 C *****
0101 115 DIF=ABS(AMAX-AMIN)
0102 C DETERMINE NUMBER OF TICK MARKS PER AXIS
0103 IF(DIF.NE.1.)GOTO 120
0104 TIC=1.
0105 GOTO 150
0106 120 IF(DIF.GT.10)GOTO 125
0107 DIF=DIF*10.
0108 GOTO 120
0109 125 IF(DIF.LE.100)GOTO 130
0110 DIF=DIF/10.
```

PAGE 0003 SCALE 9:26 AM WED., 14 JAN., 1981

```
0111      GOTO 125
0112 130   TIC=3.
0113      IF(DIF.EQ.100.)GOTO 145
0114      IF(DIF.EQ.80.)GOTO 135
0115      IF(DIF.GE.60.)GOTO 150
0116      IF(DIF.EQ.50.)GOTO 145
0117      IF(DIF.EQ.40.)GOTO 135
0118      IF(DIF.EQ.30.)GOTO 150
0119      IF(DIF.EQ.25.)GOTO 145
0120      IF(DIF.EQ.20.)GOTO 140
0121      IF(DIF.EQ.15.)GOTO 150
0122      IF(DIF.EQ.12.5)GOTO 145
0123 135   TIC=4.
0124      GOTO 150
0125 140   TIC=2.
0126      GOTO 150
0127 145   TIC=5.
0128 150   CONTINUE
0129      RETURN
0130      END
```

FTN4 COMPILER: HP92060-15092 REV. 2001 (791101)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM = 00805

COMMON = 00000

```

PAGE 0001 FTN. 1:39 PM MON., 15 DEC., 1980
0001 FTN4,L
0002 SUBROUTINE LSCAL (A,NP,PMIN,PMAX,TIC,LU,ISCAL,LERR)
0003 C LOGARITHMIC SCALING ROUTINE TO COMPUTE MAX & MIN TO LOG BASE 10
0004 C AND DETERMINE THE NUMBER OF TICK MARKS PER AXIS
0005 C A - ARRAY TO BE SCALED
0006 C NP - NUMBER OF POINTS IN ARRAY X
0007 C PMIN - POWER TO BASE 10 OF MINIMUM VALUE OF ARRAY A
0008 C PMAX - POWER TO BASE 10 OF MAXIMUM VALUE OF ARRAY A
0009 C TIC - NUMBER OF TICK MARKS ON AXIS
0010 C LU - LOGICAL UNIT NUMBER FOR ERROR OUTPUT
0011 C ISCAL - SCALING, 0 FOR AUTOMATIC, 1 FOR MANUAL
0012 C LERR - LOG ERROR, CAN'T TAKE LOG OF A NON-POSITIVE NUMBER
0013 C DIMENSION A(256)
0014 C
0015 C TEST ARRAY FOR NON-POSITIVE VALUES
0016 C
0017 LERR=0
0018 DO10 I=1,NP
0019 IF(A(I).GT.0) GOTO 10
0020 LERR=1
0021 10 CONTINUE
0022 IF(LERR.EQ.1)WRITE(LU,15)
0023 IF(LERR.EQ.1)GOTO 90
0024 IF(ISCAL.EQ.1)GOTO 25
0025 15 FORMAT(" STOP ! YOU CAN'T USE LOGARITHMIC SCALING ROUTINE (LSCAL)"
0026 -/" FOR NON-POSITIVE VALUES, USE A LINEAR SCALING ROUTINE")
0027 C
0028 C DETERMINE VALUES OF MAX AND MIN
0029 C
0030 AMIN=A(1)
0031 AMAX=AMIN
0032 DO20 I=2,NP
0033 IF(AMIN.GT.A(I))AMIN=A(I)
0034 IF(AMAX.LT.A(I))AMAX=A(I)
0035 20 CONTINUE
0036 GOTO 30
0037 25 AMIN=PMIN
0038 AMAX=PMAX
0039 C
0040 C COMPUTE LOG OF MIN AND MAX VALUES
0041 C
0042 30 POWL=ALOGT(AMIN)
0043 MINP=POWL
0044 POWH=ALOGT(AMAX)
0045 MAXP=POWH
0046 IF(ABS(FLOAT(MAXP)-POWH).LT.1.0E-05)GOTO 35
0047 IF(POWH.GE.0.0)MAXP=MAXP+1
0048 35 IF(ABS(FLOAT(MINP)-POWL).LE.1.0E-05)GOTO 40
0049 IF(POWL.GE.0)GOTO 40
0050 MINP=MINP-1
0051 40 PMIN=MINP
0052 PMAX=MAXP
0053 C MAKE DIFFERENCE IN EXPONENTS OF MAX & MIN A MULTIPLE OF 1,2,3,4 OR 5
0054 IDIF=ABS(PMAX-PMIN)
0055 IF(IDIF.LE.6)GOTO 55

```

```

PAGE 0002  LSCAL  1:39 PM MON., 15 DEC., 1980
0056      IF(IDIF.EQ.7.OR.IDIF.EQ.9)PMAX=PMAX+1
0057      IF(IDIF.LE.10)GOTO 55
0058      LDIF=15
0059  45    IF(LDIF.GE.IDIF)GOTO 50
0060      LDIF=LDIF+5
0061      GOTO 45
0062  50    PMAX=PMIN+LDIF
0063  C
0064  C DETERMINE NUMBER OF TICK MARKS PER AXIS
0065  C
0066  55    IDIF=ABS(PMAX-PMIN)
0067      IF(IDIF.NE.1)GOTO 60
0068      TIC=1
0069      GOTO 90
0070  60    IF(IDIF.GT.10)GOTO 65
0071      IDIF=IDIF*10
0072      GOTO 60
0073  65    IF(IDIF.LE.100)GOTO 70
0074      IDIF=IDIF/10
0075      GOTO 65
0076  70    TIC=3
0077      IF(IDIF.EQ.100)GOTO 85
0078      IF(IDIF.EQ.80)GOTO 75
0079      IF(IDIF.GE.60)GOTO 90
0080      IF(IDIF.EQ.50)GOTO 85
0081      IF(IDIF.EQ.40)GOTO 75
0082      IF(IDIF.EQ.30)GOTO 90
0083      IF(IDIF.EQ.25)GOTO 85
0084      IF(IDIF.EQ.20)GOTO 80
0085      IF(IDIF.EQ.15)GOTO 90
0086  75    TIC=4
0087      GOTO 90
0088  80    TIC=2
0089      GOTO 90
0090  85    TIC=5
0091  90    CONTINUE
0092      RETURN
0093      END

```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM = 00497

COMMON = 00000

```

PAGE 0001 FTN. 2:30 PM TUE., 30 DEC., 1980
0001 FTN4,L
0002 SUBROUTINE LINAX(IAXIS,AMIN,AMAX,TIC,LUG)
0003 C LINEAR AXIS DRAWING ROUTINE
0004 C
0005 C IAXIS - 1=X AXIS , 2=Y AXIS
0006 C AMIN - MINIMUM VALUE OF AXIS
0007 C AMAX - MAXIMUM VALUE OF AXIS
0008 C TIC - NUMBER OF TICK MARKS ALONG AXIS
0009 C LUG - LOGICAL UNIT NUMBER FOR GRAPHICS OUTPUT
0010 C
0011 COMMON IGCB(192)
0012 C SET AXIS LENGTH IN WORLD COORDINATE SYSTEM (WCS)
0013 IF(IAXIS.EQ.1)ALEN=114.
0014 IF(IAXIS.EQ.2)ALEN=70.
0015 C DEFINE ORIGIN
0016 XO=19.
0017 YO=10.
0018 TICM=AMIN
0019 CALL MOVE(IGCB,XO,YO)
0020 CALL CSIZE(IGCB,3.)
0021 IF(IAXIS.EQ.2)GOTO 100
0022 C
0023 C DRAW X AXIS
0024 C
0025 CALL DRAW(IGCB,ALEN+XO,YO)
0026 C LABEL X ORIGIN
0027 CALL MOVE(IGCB,XO,YO)
0028 CALL MOVEI(IGCB,-12.,-4.5)
0029 CALL LABEL(IGCB)
0030 WRITE(LUG,30)TICM
0031 30 FORMAT(F8.0)
0032 C DRAW X TICK MARKS
0033 DO 50 K=1,TIC
0034 TICK=ALEN*(FLOAT(K)/TIC)
0035 CALL MOVE(IGCB,TICK+XO,YO)
0036 CALL DRAW(IGCB,TICK+XO,YO+2.0)
0037 C LABEL X TICK MARKS
0038 TICM=TICM+((AMAX-AMIN)/TIC)
0039 CALL MOVE(IGCB,TICK+XO,YO)
0040 CALL MOVEI(IGCB,-12.,-4.5)
0041 CALL LABEL(IGCB)
0042 WRITE(LUG,30)TICM
0043 50 CONTINUE
0044 GOTO 200
0045 C
0046 C DRAW Y AXIS
0047 C
0048 100 CALL DRAW(IGCB,XO,ALEN+YO)
0049 C LABEL Y ORIGIN
0050 CALL MOVE(IGCB,XO,YO)
0051 CALL MOVEI(IGCB,-17.,-0.8)
0052 CALL LABEL(IGCB)
0053 WRITE(LUG,30)TICM
0054 C DRAW Y TICK MARKS
0055 DO 150 K=1,TIC

```

PAGE 0002 LINAX 2:36 PM TUE., 30 DEC., 1980

```
0056      TICK=ALEN*(FLOAT(K)/TIC)
0057 C LABEL Y TICK MARKS
0058      TICM=TICM+((AMAX-AMIN)/TIC)
0059      CALL MOVE(IGCB,XO,TICK+YO)
0060      CALL DRAW(IGCB,XO+2.5,TICK+YO)
0061      CALL MOVE(IGCB,XO,TICK+YO)
0062      CALL MOVEI(IGCB,-17.,-0.8)
0063      CALL LABEL(IGCB)
0064      WRITE(LUG,30)TICM
0065 150  CONTINUE
0066 200  CALL PENUP(IGCB)
0067      RETURN
0068      END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM = 00342

COMMON = 00192

```

PAGE 0001 FTN. 2:23 PM TUE., 2 DEC., 1980
0001 FTN4,L
0002 SUBROUTINE LOGAX(IAxis,PMIN,PMAX,TIC,LUG)
0003 C LOGRATHMETIC AXIS DRAWING ROUTINE
0004 C
0005 C IAXIS - 1=X AXIS , 2=Y AXIS
0006 C PMIN - POWER TO BASE 10 OF MINIMUM VALUE OF AXIS
0007 C PMAX - POWER TO BASE 10 OF MAXIMUM VALUE OF AXIS
0008 C TIC - NUMBER OF TICK MARKS ALONG AXIS
0009 C LUG - LOGICAL UNIT NUMBER FOR GRAPHICS OUTPUT
0010 C
0011 COMMON IGCB(192)
0012 C SET AXIS LENGTH IN WORLD COORDINATE SYSTEM (WCS)
0013 IF(IAxis.EQ.1)ALEN=114.
0014 IF(IAxis.EQ.2)ALEN=70.
0015 C DEFINE ORIGIN
0016 XO=19.
0017 YO=10.
0018 IEXP=PMIN
0019 CALL CSIZE(IGCB,3.)
0020 CALL MOVE(IGCB,XO,YO)
0021 IF(IAxis.EQ.2)GOTO 25
0022 C
0023 C DRAW X AXIS
0024 C
0025 CALL DRAW(IGCB,ALEN+XO,YO)
0026 C LABEL X ORIGIN
0027 CALL MOVE(IGCB,XO,YO)
0028 CALL MOVEI(IGCB,-3.7,-4.5)
0029 CALL LABEL(IGCB)
0030 WRITE(LUG,10)
0031 10 FORMAT("10")
0032 CALL CSIZE(IGCB,2.)
0033 CALL MOVEI(IGCB,2.5,2.)
0034 CALL LABEL(IGCB)
0035 WRITE(LUG,15)IEXP
0036 15 FORMAT(I3)
0037 CALL CSIZE(IGCB,3.)
0038 C DRAW X TICK MARKS
0039 DO20 K=1,TIC
0040 TICK=ALEN*(FLOAT(K)/TIC)
0041 CALL MOVE(IGCB,TICK+XO,YO)
0042 CALL DRAW(IGCB,TICK+XO,YO+2.0)
0043 C LABEL X TICK MARKS
0044 CALL MOVE(IGCB,TICK+XO,YO)
0045 CALL MOVEI(IGCB,-3.7,-4.5)
0046 CALL LABEL(IGCB)
0047 WRITE(LUG,10)
0048 CALL CSIZE(IGCB,2.)
0049 CALL MOVEI(IGCB,2.5,2.)
0050 IEXP=IEXP+(PMAX-PMIN)/TIC
0051 CALL LABEL(IGCB)
0052 WRITE(LUG,15)IEXP
0053 CALL CSIZE(IGCB,3.)
0054 20 CONTINUE
0055 GOTO 35

```

PAGE 0002 LOGAX 2:23 PM TUE., 2 DEC., 1980

```
0056 C
0057 C DRAW Y AXIS
0058 C
0059 25 CALL DRAW(IGCB,XO,ALEN+YO)
0060 C LABEL Y ORIGIN
0061 CALL MOVE(IGCB,XO,YO)
0062 CALL MOVEI(IGCB,-9.5,-1.5)
0063 CALL LABEL(IGCB)
0064 WRITE(LUG,10)
0065 CALL CSIZE(IGCB,2.)
0066 CALL MOVEI(IGCB,2.5,2.)
0067 CALL LABEL(IGCB)
0068 WRITE(LUG,15)PMIN
0069 CALL CSIZE(IGCB,3.)
0070 C DRAW Y TICK MARKS
0071 DO30 K=1,TIC
0072 TICK=ALEN*(FLOAT(K)/TIC)
0073 C LABEL Y TICK MARKS
0074 CALL MOVE(IGCB,XO,TICK+YO)
0075 CALL DRAW(IGCB,XO+2.5,TICK+YO)
0076 CALL MOVE(IGCB,XO,TICK+YO)
0077 CALL MOVEI(IGCB,-9.5,-1.5)
0078 CALL LABEL(IGCB)
0079 WRITE(LUG,10)
0080 CALL CSIZE(IGCB,2.)
0081 CALL MOVEI(IGCB,2.5,2.)
0082 IEXP=IEXP+(PMAX-PMIN)/TIC
0083 CALL LABEL(IGCB)
0084 WRITE(LUG,15)IEXP
0085 CALL CSIZE(IGCB,3.)
0086 30 CONTINUE
0087 35 CALL PENUP(IGCB)
0088 RETURN
0089 END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM = 00439

COMMON = 00192

```

PAGE 0001 FTN. 1:44 PM TUE., 2 DEC., 1980
0001 FTN4,L
0002 SUBROUTINE LSREG (X,Y,IFRM,ITO,SLOPE,YINT,SDSLOP,SDYINT)
0003 C PROGRAM CALCULATES LEAST SQUARES REGRESSION
0004 C X - X ARRAY
0005 C Y - Y ARRAY
0006 C IFRM - STARTING POINT OF INTERVAL
0007 C ITO - ENDING POINT OF INTERVAL
0008 C SLOPE - RISE OVER RUN OF L.S. LINE
0009 C YINT - Y INTERCEPT OF L.S. LINE
0010 C SDSLOP - STANDARD DEVIATION OF SLOPE
0011 C SDYINT - STANDARD DEVIATION OF Y INTERCEPT
0012 DIMENSION X(256),Y(256)
0013 FN=ITO-IFRM+1
0014 TX=0
0015 ZY=0
0016 XY=0
0017 WY=0
0018 SY=0
0019 DO 100 K=IFRM,ITO
0020 SY=SY+(Y(K)**2)
0021 WY=WY+(X(K)*Y(K))
0022 XY=XY+X(K)
0023 ZY=ZY+Y(K)
0024 100 TX=TX+(X(K)**2)
0025 TY=(ZY**2)/FN
0026 XZY=(XY*ZY)/FN
0027 XYN=(WY-XZY)**2
0028 XYD=TX-((XY**2)/FN)
0029 SX=(XY**2)/FN
0030 SDSLOP=SQRT((SY-TY-(XYN/XYD))/((FN-2.)*(TX-SX)))
0031 SDYINT=SQRT((SDSLOP**2)*(TX/FN))
0032 SLOPE=((FN*WY)-(XY*ZY))/((FN*TX)-(XY**2))
0033 YINT=(ZY-(SLOPE*XY))/FN
0034 RETURN
0035 END
FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)
** NO WARNINGS ** NO ERRORS ** PROGRAM = 00315 COMMON = 00000

```

DATE  
FILMED  
-8